Math

8

UIUCDCS-R-77-835

UILU-ENG 77 1720

PERFORMANCE ANALYSIS OF MULTIPROCESSOR SYSTEMS CONTAINING FUNCTIONAL DEDICATED PROCESSORS

by

Jane W. S. Liu and Chung L. Liu

May 1977

DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN · URBANA, ILLINOIS

UIUCDCS-R-77-835

PERFORMANCE ANALYSIS OF MULTIPROCESSOR SYSTEMS CONTAINING FUNCTIONAL DEDICATED PROCESSORS

by

Jane W. S. Liu

Chung L. Liu

May 1977

Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, Illinois 61801

This work was supported by the National Science Foundation under Grant NSFDCR 72-03740 NSFMCS 73-03408.

I. INTRODUCTION

In previous works on job scheduling, a multiprocessor system was modeled as one containing functionally identical processors. A task in a given set of tasks can be assigned to any one of the processors in the system. The time required to complete a task may depend only on the execution time of the task in the case where all processors are identical or may also depend on the speed of the processor on which the task is being processed in the case where processors have different speeds. For such multiprocessor systems, efficient algorithms which yield schedules with minimum completion time or mean flow times have been found for some special cases [1-5]. Worst case performance bounds of many heuristic suboptimal scheduling algorithms have also been obtained [6-8].

In this report, we introduce two general models of multiprocessor systems containing different types of processors. A task can be assigned only to a processor of certain types. We described in Section II a model corresponding to systems in which processors are functionally dedicated [9]. That is, a task can be executed only on a particular type of processors. (For example, some processors are front end processors for I/O functions, some processors are to perform program compilation, some processors are designed specially for merging/sorting operations, and others are designed for extensive numerical calculations.) Clearly, the model of multiprocessor systems with identical processors is a special case of our model. This model also includes the job shop problem in which there is exactly one processor of each type.

In Section III, a more general model is presented. In this general model, each type of functionally dedicated processors is further

divided into subtypes with a partial ordering relation defined over the processor subtypes. Thus, multiprocessor systems in which some processors are functionally identical but have dedicated memories of different sizes can be modeled.

II. MULTIPROCESSOR SYSTEMS WITH FUNCTIONALLY DEDICATED PROCESSORS

Consider a model of a multiprocessor system in which there are r different types of functionally dedicated processors. We assume in this section that processors of the same type are identical. We shall refer to these r types of processors as type 1, type 2, ... , type r processors. A multiprocessor system with m_1 type 1 processors, m_2 type 2 processors, ... , and m_r type r processors can be represented by the ordered r-tuple $P = (m_1, m_2, \ldots, m_r)$. We let $m = \sum_{j=1}^{r} m_j$.

Let $T = \{T_1, T_2, \ldots, T_n\}$ be a set of tasks to be executed on a system P. A task T_i is said to be a type j task if it can only be executed on a type j processor. Formally, we define a function λ from T to the set $\{1, 2, \ldots, r\}$ such that $\lambda(T_i)$ is the type of task T_i. We denote the time required to complete a task T_i (on a type j processor) by $\mu(T_i)$ where μ is a function from T to the reals. $\mu(T_i)$ shall be referred to as the execution time of T_i.

We suppose that there is a precedence relation $<$ defined over the set T. That T_i precedes T_j (or T_j follows T_i) is written as $T_i < T_j$ and means that the execution of T_j cannot begin before the execution of T_i is completed. A task is said to be executable at a certain time if the tasks preceding it have been completed. Formally, a set of tasks are represented by ordered quadruple $(T, \lambda, \mu, <)$. We shall also use the notation $(T_i, \lambda(T_i), \mu(T_i))$ to represent to a particular task T_i.

We can depict a set of Tasks $(T, \lambda, \mu, <)$ graphically as illustrated by the example in Figure 1, where T_1, T_2, and T_5 are type 1 tasks with execution time 3, 4 and 2, respectively. T_3 and T_4 are type 2 tasks with execution time 4 and 2, respectively.

We want to determine the performance of a class of scheduling

algorithms in which processors are never left idle intentionally. These algorithms are known as priority driven scheduling algorithms and can be

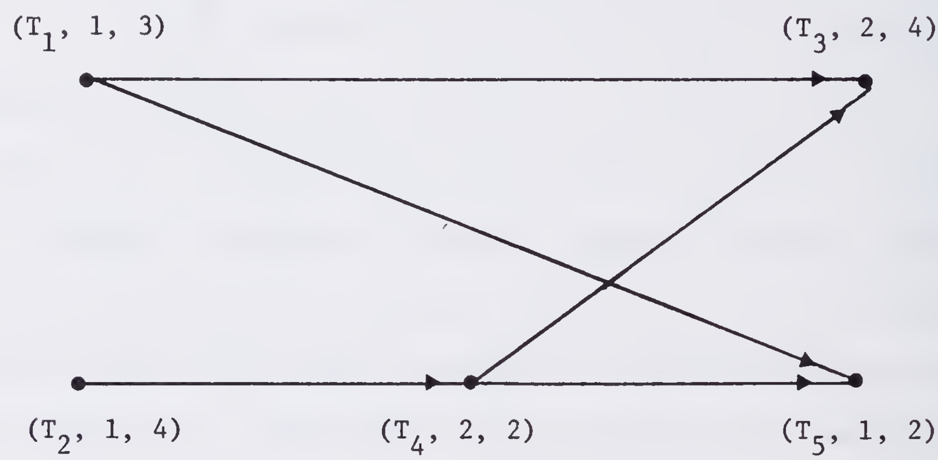


Figure 1. An example of a Set of Tasks ($\mathcal{T}$, λ, μ, $<$).

described by the priorities assigned to the tasks. At any moment when a type i processor is free, the task that has the highest priority among all executable type i tasks is scheduled. For example, for the set of tasks in Figure 2(a), the schedule in Figure 2 (b) is obtained according to the priority list (T_1, T_2, T_3, T_4, T_5, T_6, T_7, T_8, T_9) in which tasks appear in decreasing priorities. Schedules produced by priority driven scheduling algorithms are known as priority driven schedules.

Let ω and ω' denote the completion times of a set of tasks $(\mathcal{T}, \lambda, \mu, <)$ when executed on a system $P = (m_1, m_2, \dots, m_r)$ according to a priority-driven schedule and an arbitrary schedule, respectively. We have

<u>Theorem 1</u>

$$\frac{\omega}{\omega'} \leq 1 + r - \min_j \left(\frac{1}{m_j}\right) \tag{1}$$

Moreover, the bound is the best possible.

<u>Proof</u>: Let t_j denote the sum of the execution times of all type j tasks in $(\mathcal{T}, \lambda, \mu, <)$. (For example, for the set of tasks in Figure 2, $t_1 = 5\frac{1}{2}$ and $t_2 = 3\frac{1}{2}$). Let $\phi(n_1, n_2, \dots, n_r)$ denote the sum of idle times in all processors during which there are n_1 idle processors of type 1, n_2 idle processors of type 2, ... , and n_r idle processors of type r in a priority-driven schedule. (In the example shown in Figure 2, $\phi(1,0) = 1$, $\phi(2,0) = \frac{1}{2} + \frac{1}{2} = 1$, $\phi(1,1) = 2 + 2 = 4$, $\phi(0,0) = 0$, $\phi(0,1) = 0$, and $\phi(2,1) = 0$.) Clearly, the total idle time, Φ, is given by

$$\Phi = \sum_{n_1=0}^{m_1} \sum_{n_2=0}^{m_2} \cdots \sum_{n_r=0}^{m_r} \phi(n_1, n_2, \dots, n_r) \tag{2}$$

with $\phi(m_1, m_2, \dots, m_r) = 0$.[†] The completion time of the priority-

---

[†]Note that $\phi(m_1, m_2, \dots, m_r) = 0$ in a priority-driven schedule.

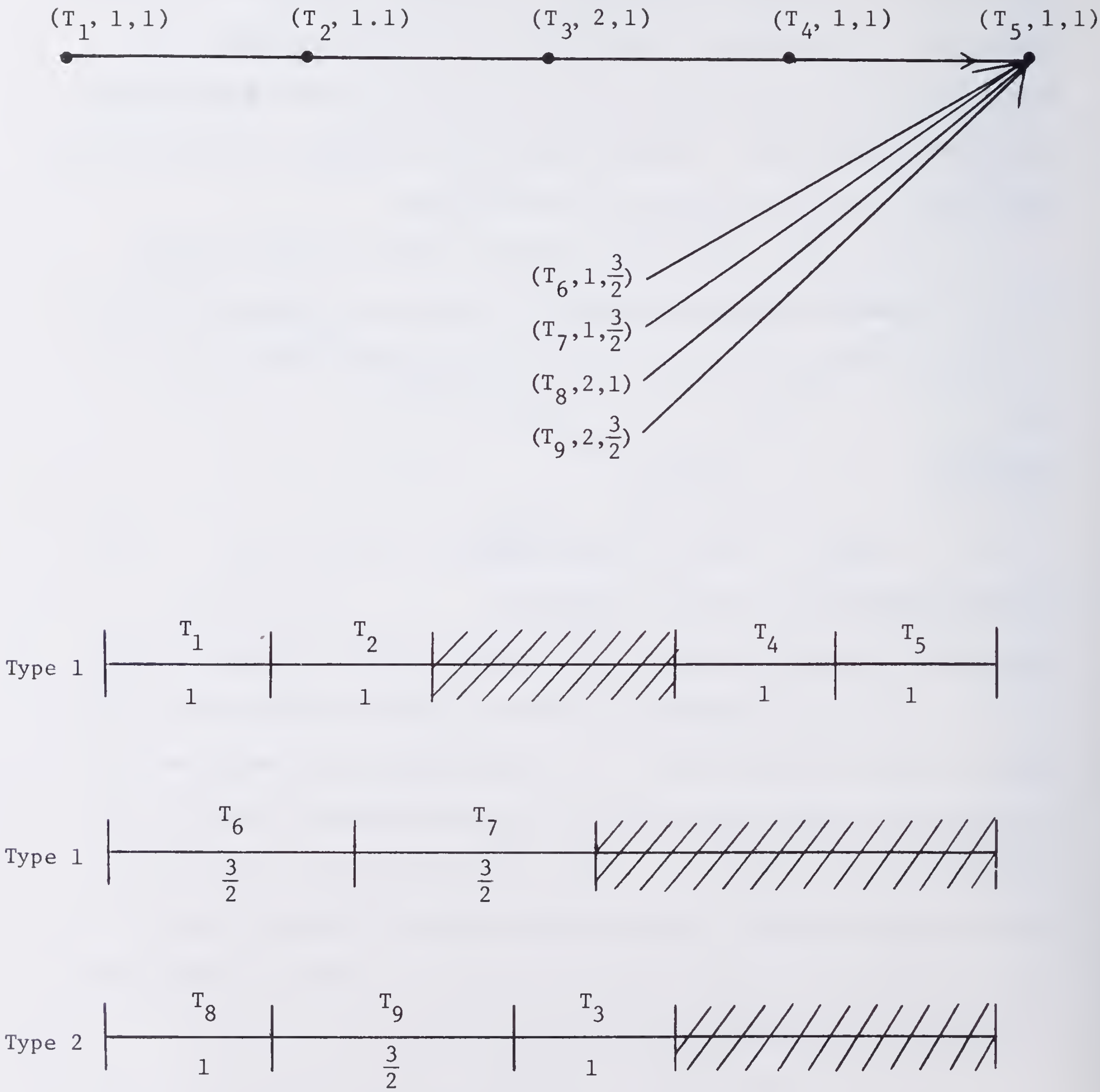


Figure 2. Example of a priority driven schedule on a multiprocessor system containing different types of processors.

driven schedule ω is given by

$$\omega = \frac{1}{m} \left(\sum_{j=1}^{r} t_j + \Phi \right) \tag{3}$$

To find an upper bound to the total idle time, Φ, let us examine the terms

$$I = \sum_{n_1=1}^{m_1} \sum_{n_2=1}^{m_2} \cdots \sum_{n_r=1}^{m_r} \phi(n_1, n_2, \ldots n_r)$$

in the sum in Eq. (2). During the time periods corresponding to the terms in the sum I, there is at least one idle processor among the processors of each type. Therefore, there is a chain of tasks in T that is executed sequentially during these idle periods, and must be executed sequentially in any schedule. Thus, we conclude that

$$I \leq (m-1)\omega' \tag{4}$$

since no schedule can have a completion time ω' less than the length of this chain.

Let K_j denote the sum of lengths of idle periods during which all type j processors are busy. We have

$$K_j \leq \frac{m-m_j}{m_j} \left[t_j - \sum_{\substack{1 \leq n_j \leq m_j - 1 \\ 0 \leq n_u \leq m_u \text{ for } u \neq j}} \frac{m_j - n_j}{n_1 + n_2 + \ldots + n_r} \phi(n_1, n_2, \ldots, n_r) \right]$$

$$\leq \frac{m-m_j}{m_j} t_j - \frac{m-m_j}{m_j} \cdot \frac{1}{m-1} \sum_{\substack{1 \leq n_j \leq m_j - 1 \\ 0 \leq n_u \leq m_u \text{ for } u \neq j}} \phi(n_1, n_2, \ldots, n_r)$$

Furthermore,

$$\sum_{j=1}^{r} K_j \leq \sum_{j=1}^{r} \frac{m-m_j}{m_j} t_j - \sum_{j=1}^{r} \frac{m-m_j}{m_j} \cdot \frac{1}{m-1} \sum_{\substack{1 \leq n_j \leq m_j - 1 \\ 0 \leq n_u \leq m_u \text{ for } u \neq j}} \phi(n_1, n_2, \ldots, n_r)$$

$$\leq \sum_{j=1}^{r} \frac{m-m_j}{m_j} t_j - \frac{I}{m-1} \min_{1\leq j\leq r} \frac{m-m_j}{m_j} \tag{5}$$

Since

$$\Phi \leq I + \sum_{j=1}^{r} K_j$$

we have from Eq. (5),

$$\Phi \leq \sum_{j=1}^{r} \frac{m-m_j}{m_j} t_j + I\left[1 - \frac{1}{m-1} \min_{1\leq j\leq r} \frac{m-m_j}{m_j}\right].$$

Combining this expression with the inequality in Eq. (4) and that

$$\frac{t_j}{m_j} \leq \omega' \qquad j = 1, 2, \ldots , r.$$

We have

$$\Phi \leq m(r-1)\omega' + m\omega' \left(1 - \min_{1\leq j\leq r} \frac{1}{m_j}\right) \tag{6}$$

Furthermore, since

$$\frac{1}{m} \sum_{j=1}^{r} t_j \leq \omega'. \tag{7}$$

We have the inequality Eq.(1) by substituting Eqs. (6) and (7) into Eq. (3).

The example shown in Figure 3 demonstrates that the upper bound in Eq. (1) is best possible. In this example, we have

$$m_1 = \max_{1\leq j\leq r} m_j$$

There are $m_1(m_1-1) + 2$ type 1 tasks which are denoted as $T_{11}, T_{12}, \ldots , T_{1,(m_1(m_1-1)+2)}$. Their execution times are

$$\mu(T_{11}) = \varepsilon$$

$$\mu(T_{1i}) = \frac{1}{m_1} \qquad i = 2, 3, \ldots , m_1(m_1-1) + 1$$

$$\mu(T_{1,m_1(m_1-1)+2}) = 1$$

There are $m_j + 1$ type j tasks for $j = 2, 3, \ldots , r$ which we shall denote as

$T_{j1}, T_{j2}, \cdots, T_{j,(m_j+1)}$ for j = 2, 3, ... r. Their execution times are

$$\mu(T_{j\ell}) = 1 \qquad j = 2, 3, \ldots, r$$

and $\qquad \ell = 1, 2, \ldots, m_j$

$$\mu(T_{j,(m_j+1)}) = \varepsilon \quad j = 2, 3, \ldots, r$$

Furthermore, the precedence relation is as shown in Figure 3a. The priority schedule in Figure 3b is obtained according to the priority list

$$\Big(T_{12}, T_{13}, \cdots, T_{1,(m_1(m_1-1)+1)}, T_{11}, T_{21}, T_{22} \cdots T_{2,(m_2+1)}, T_{31}, T_{32}, \\ \cdots T_{r,(m_r+1)}, T_{1,(m_1(m_1-1)+2)}\Big)$$

It's complete time is

$$\omega = r + \frac{m_1-1}{m_1} + r\varepsilon$$

while the completion time of the schedule shown in Figure 3c is

$$\omega' = 1 + r\varepsilon$$

Hence, the bound in Eq. (1) is achieved for very small ε

When the processors in the system are all identical, r = 1. The result in Theorem 1 reduces to the well-known result [6]:

<u>Corollary 1</u>: For a system containing m identical processors

$$\frac{\omega}{\omega'} \le 2 - \frac{1}{m}$$

On the other hand, for a job shop problem, $m_1 = m_2 = \ldots = m_r = 1$

In this case, we have,

<u>Corollary 2</u>: For a r-processor job shop

$$\frac{\omega}{\omega'} \le r.$$

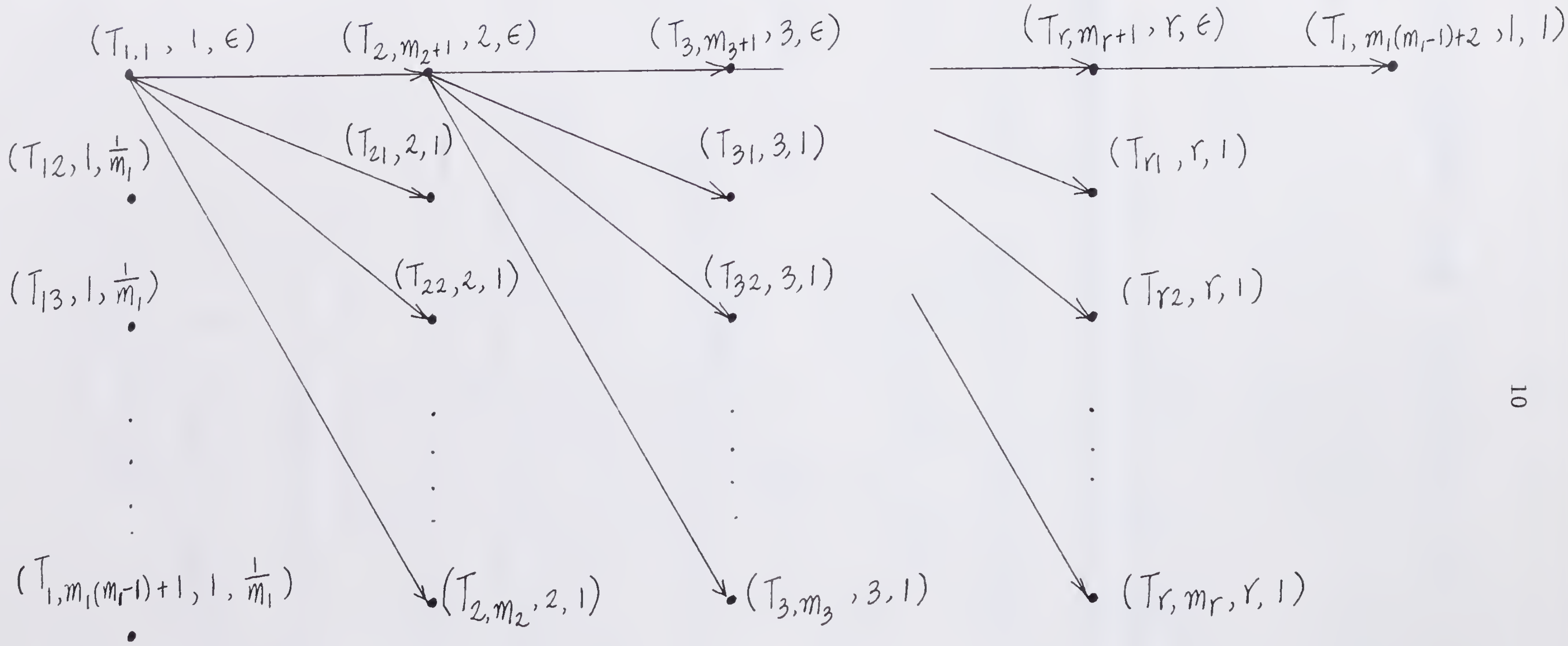
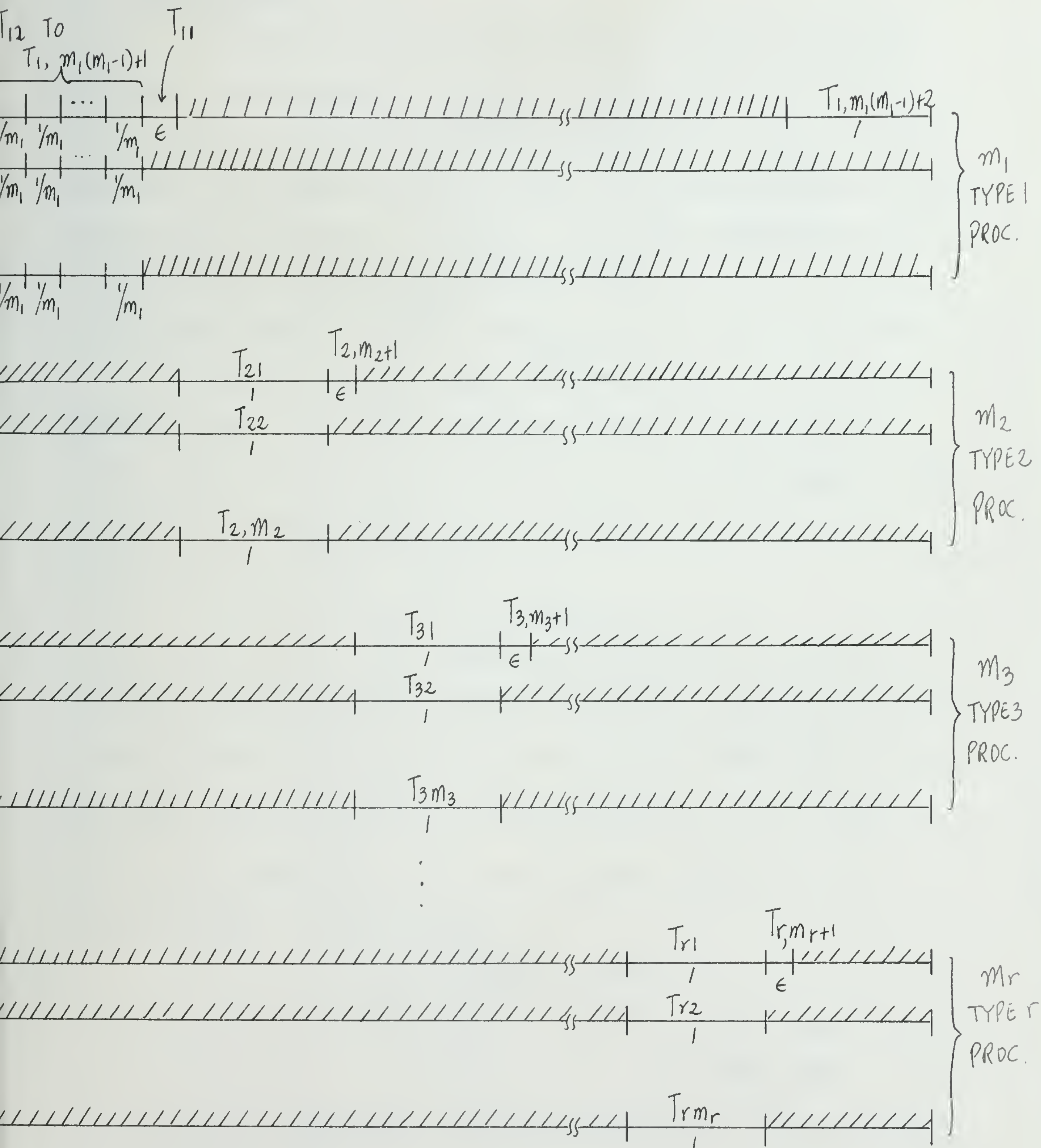
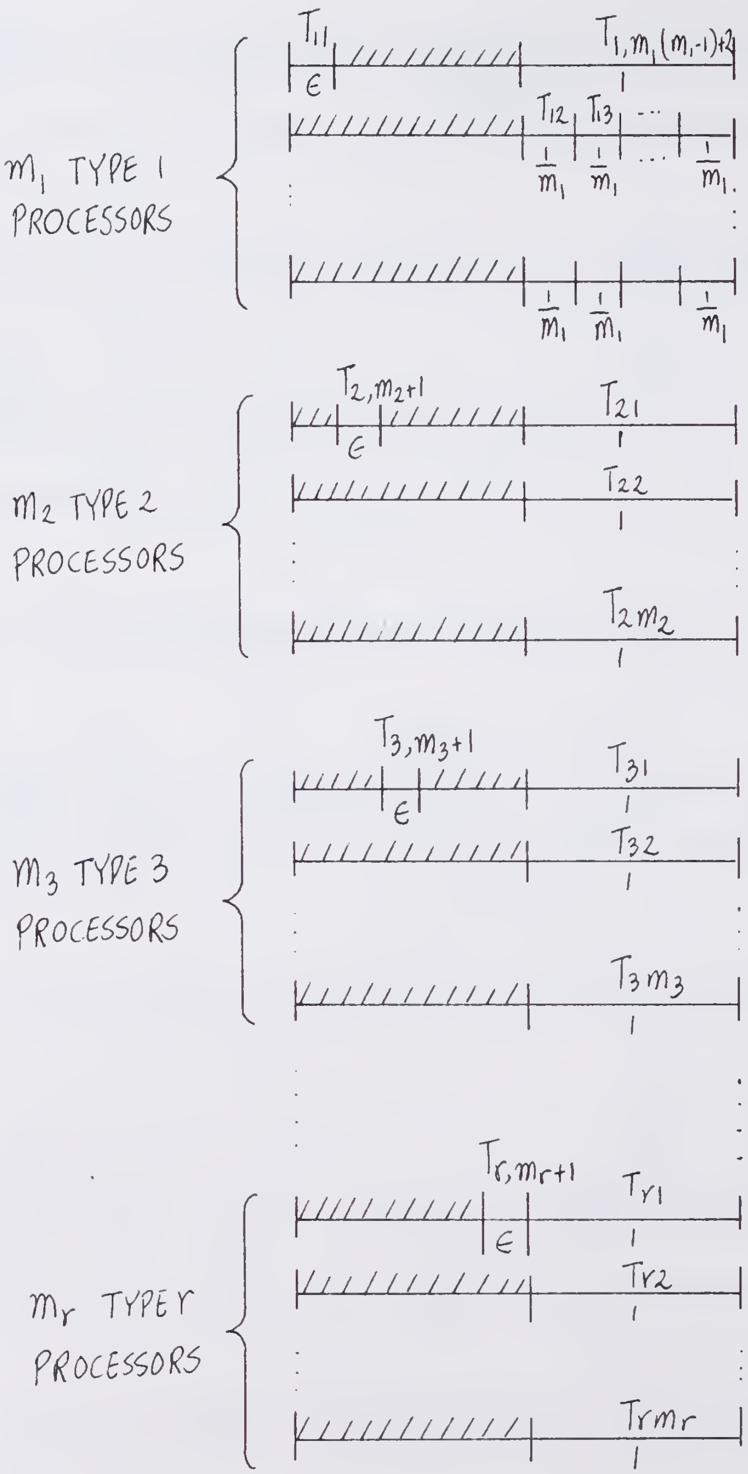


Figure 3a. Precedence Graph of Example 1.



$$\omega = r + \frac{m_1 - 1}{m_1} + r\,\varepsilon$$

Figure 3b. Worst Case Priority-Driven Schedule.



$\omega = 1 + r\,\varepsilon$

Figure 3c. Optimal Schedule.

III. GENERALIZATION

The model of a multiprocessor system in which each processor has a dedicated memory of a certain fixed capacity was studied in [8]. In this case, the execution of a task requires a certain amount of memory space and can only take place on a processor whose memory capacity is larger than or equal to its memory requirement. Our model in Section II can be extended such that every processor is characterized by its type as well as by its memory capacity. We shall, however, present a more general model.

Consider a multiprocessor system consisting of processors of r different types. Each type of processors is further divided into subtypes. Thus, we shall refer to a processor as a type (j,k) processor when it is a type j processor of subtype k. A partial ordering relation $\prec$ is defined over the processor types such that

(i) (j,k) and (n,v) are incomparable† if $j \neq n$.

(ii) $(j,k) \prec (j,v)$ means that if a task can be executed on a type (j,k) processor then it can also be executed on a type (j,v) processor.

A multiprocessor system can then be represented as $P = (m_{11}, m_{12},\ldots,m_{1\ell_1}; m_{21}, m_{22},\ldots,m_{2\ell_2};\ldots m_{r1}, m_{r2}, \ldots, m_{r\ell_r}; \prec)$ where m_{jk} is the number of type (j,k) processors and $\prec$ is a partial ordering relation over the processor types. We shall let

$$m_j = \sum_{k=1}^{\ell_j} m_{jk}$$

$$m = \sum_{j=1}^{r} m_j$$

For example, the multiprocessor system represented by the directed graph†

---

† Neither $(j,k) \prec (u,v)$ nor $(u,v) \prec (j,k)$.

† There is an edge from (j,k) to (j,v) means $(j,v) \prec (j,k)$.

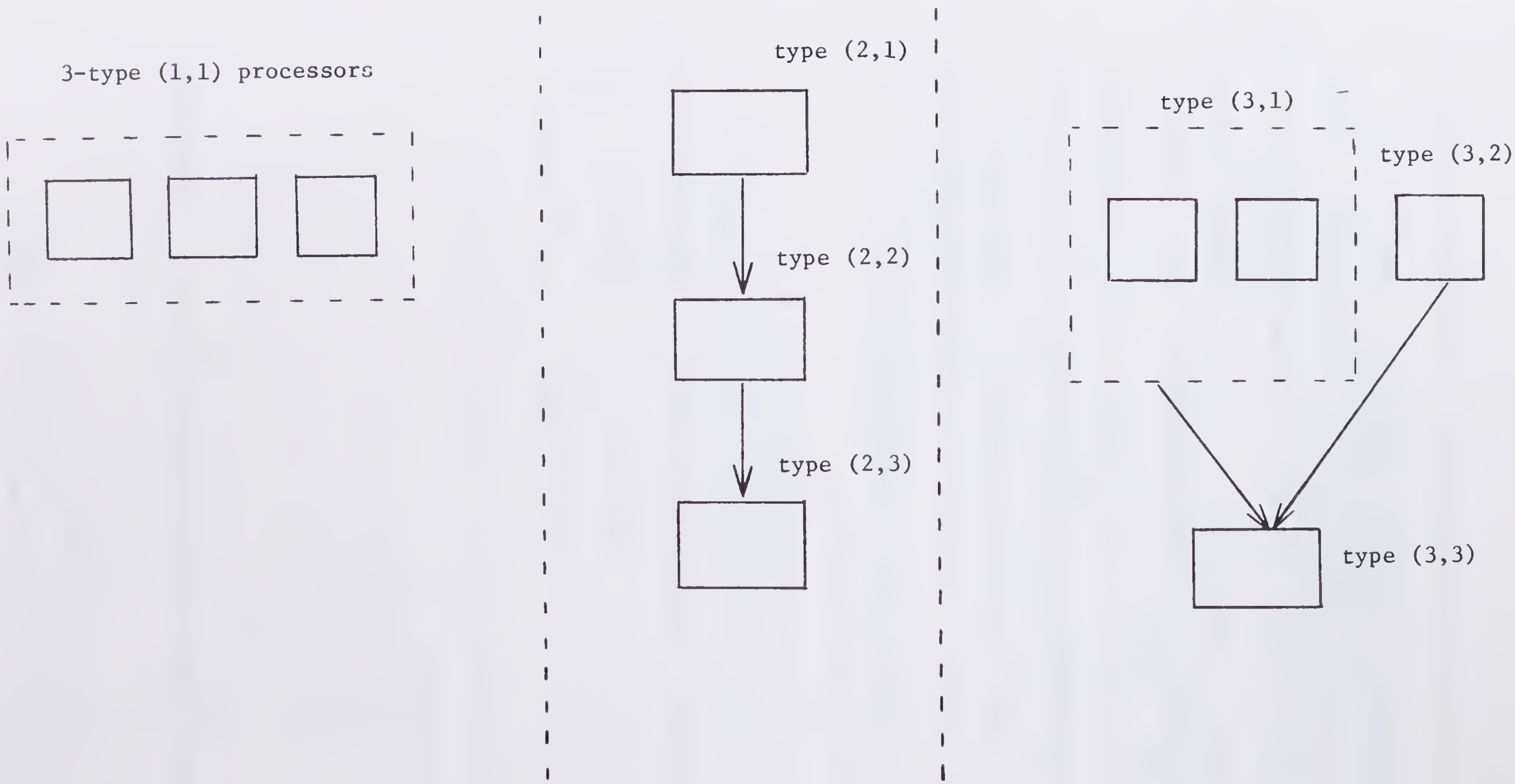

3-type (1,1) processors
type (2,1)
type (2,2)
type (2,3)
type (3,1)
type (3,2)
type (3,3)

Figure 4

in Figure 4 contains three types of processors. There are three type 1 processors of the same subtype. The 3 processors of type 2 having dedicated memories of different sizes are linearly ordered as shown. For the 4 processors of type 3, neither (3.1) $\prec$ (3,2) nor (3,2) $\prec$ (3.1) but (3,3) $\prec$ (3,1) and (3,3) $\prec$ (3,2). This system can be represented as $\mathcal{P}$=(3;1,1,1;2,1,1,).

Let $\mathcal{T} = \{T_1, T_2, \ldots, T_n\}$ be a set of tasks to be exectued on a system $\mathcal{P}$. A task T_i is said to be a type (j,k) task if it can be executed on any type (j,v) processor for (j,k) $\prec$ (j,v) and on no other type of processors. Thus, a set of tasks can be specified by an ordered 4-tuple ($\mathcal{T}$, λ, μ, <) where λ is a function from T_i to the processor types so that $\lambda(T_i)$ specifies the type of T_i, and μ and < are as defined in Section II. Again, we use the notation $(T_i, \lambda(T_i), \mu(T_i))$ to represent a particular task T_i.

Consider all type (j,k) processors for a fixed j. The smallest subset of types $\{(j,k_1), (j,k_2), \ldots (j,k_q)\}$ is said to be the dominating set if for any type (j,k), $(j,k) \prec (j,k_p)$ for some $1 \leq p \leq q$. In other words, any type (j,k) task can be executed on a processor whose type belongs to the dominating set. For example, in the multiprocessor system shown in Figure 4, {(1,1)}, {(2,1)} and {(3,1), (3,2)} are dominating sets. We also refer to a type (j,k) as a maximal type if it is in the dominating set.

For a given dominating set, let

$$m_{j0} = \min\{m_{jk_p} \mid (j,k_p) \text{ is in the dominating set}\}$$

In other words, m_{j0} is the minimum of the numbers of processors among all types of processors in the dominating set. Therefore, m_{j0} is equal to 3, 1, and 1 for j = 1, 2, and 3, respectively, in our example. The result in Theorem 1 can be generalized to

<u>Theorem 2</u>

$$\frac{\omega}{\omega'} \leq 1 + \sum_{j=1}^{r} \frac{m_j}{m_{j0}} - \min_{1 \leq j \leq r} \frac{1}{m_{j0}} \qquad (8)$$

Proof. Again, let t_j denote the total execution times of all type (j,k) tasks for $k = 1, 2, \ldots, \ell_j$. For a priority-driven schedule, let $\phi(n_{11}, n_{12}, \ldots, n_{1\ell_1}, n_{21}, \ldots, n_{2\ell_2}, \ldots, n_{r1}, \ldots, n_{r\ell_r})$ denote the sum of idle times in all processors in P during which there are n_{jk} idle processors of type (j,k) for $j = 1, 2, \ldots r$ and $k = 1, 2, \ldots, \ell_j$. And the total idle time

$$\Phi = \sum_{n_{11}=0}^{m_{11}} \sum_{n_{12}=0}^{m_{12}} \cdots \sum_{n_{r\ell_r}=0}^{m_{r\ell_r}} \phi(n_{11}, n_{12}, \ldots, n_{r\ell_r}) \qquad (9)$$

Hence the completion time of the priority driven schedule is

$$\omega = \frac{1}{m}\left[\sum_{j=1}^{r} t_j + \Phi\right]$$

as given by Eq. (3).

Let I be the sum of all terms in Eq. (9) corresponding to idle periods during which at least one of each maximal type processor is idle. That is

$$I = \sum_{j=1}^{r} \sum_{k=1}^{\ell_j} \sum_{n_{jk} \in S_{jk}} \phi(n_{11}, n_{12}, \ldots, n_{r\ell_r})$$

where $S_{jk} = \{1, 2, \ldots, m_{jk} - 1\}$ if (j,k) is a maximal type and, $S_{jk} = \{0, 1, 2, \ldots, m_{jk}\}$ if (j,k) is not a maximal type.

Again, there is a chain of task in T such that during these idle periods one of the other processors is executing a task in the chain. Hence

$$I \leq (m-1)\omega' \qquad (10)$$

Let K_j denote the sum of lengths of idle periods during which all the processors of one or more maximal types of the form $(j,*)$ are busy. We have

$$K_j \leq \frac{m-m_{j0}}{m_{j0}}\left[t_i - \sum_{j=1}^{r}\sum_{k=1}^{\ell_j}\sum_{n_{jk}\in S_{jk}} \frac{m_j - n_{j1} - j_{j2} - \cdots - n_{j\ell_j}}{n_{11}+n_{12}+\cdots+n_{r\ell_r}}\,\phi(n_{11}, n_{12}, \cdots n_{r\ell_r})\right]$$

$$\leq \frac{m-m_{j0}}{m_{j0}}\left[t_i - \frac{1}{m-1}\sum_{j=1}^{r}\sum_{k=1}^{\ell_j}\sum_{n_{jk}\in S_{jk}} \phi(n_{11}, n_{12}, \cdots n_{r\ell_r})\right]$$

Hence

$$\sum_{j=1}^{r} K_j \leq \sum_{j=1}^{r} \frac{m-m_{j0}}{m_{j0}}\, t_j - \frac{I}{m-1} \min_{1\leq j\leq r} \frac{m-m_{j0}}{m_{j0}}$$

and

$$\Phi \leq I + \sum_{j=1}^{r} K_j$$

$$\leq \sum_{j=1}^{r} \frac{m-m_{j0}}{m_{j0}}\, t_j + I\left[1 - \frac{1}{m-1}\min_{1\leq j\leq r}\frac{m-m_{j0}}{m_{j0}}\right]$$

Since

$$\frac{t_j}{m_j} \leq \omega' \qquad j = 1, 2, \ldots, r$$

and

$$\frac{1}{m}\sum_{j=1}^{r} t_j \leq \omega'$$

as well as Eq. (10), we obtain

$$\omega \leq \omega' + \frac{\omega'}{m}\left\{\sum_{j=1}^{r}(m-m_{j0})\frac{m_j}{m_{j0}} + (m-1)\left[\omega - \frac{1}{m-1}\min_{1\leq j\leq r}\frac{m-m_{j0}}{m_{j0}}\right]\right\}$$

$$= \omega'\left[1 + \sum_{j=1}^{r}\frac{m_j}{m_{j0}} - \frac{1}{m}\sum_{j=1}^{r} m_j + 1 - \min_{1\leq j\leq r}\frac{1}{m_{j0}}\right]$$

$$= \omega' \left[1 + \sum_{j=1}^{r} \frac{m_j}{m_{j0}} - \min_{1 \le j \le r} \frac{1}{m_{j0}} \right]$$

which is the bound given by Eq. (8). When the dominating sets contains only one subtype for all $j = 1, 2, \ldots, r$, (that is, there is an unique maximal subtype for all types), we have

Corollary 3: The bound given by (8) is the best possible.

That this bound is best is illustrated by the example in Figure 5.

When the multiprocessor system contains only one type of processors, we have $m_i = 0$ for $i = 2, 3, \ldots, r$. The bound given by Theorem 2 is simply

Corollary 4:

$$\frac{\omega}{\omega'} \le 1 + \frac{m_1}{m_{11}} - \frac{1}{m_{11}}$$

The bound derived in [8] for processors with different storage capacities is a special case of our result with $m_{11} = 1$.

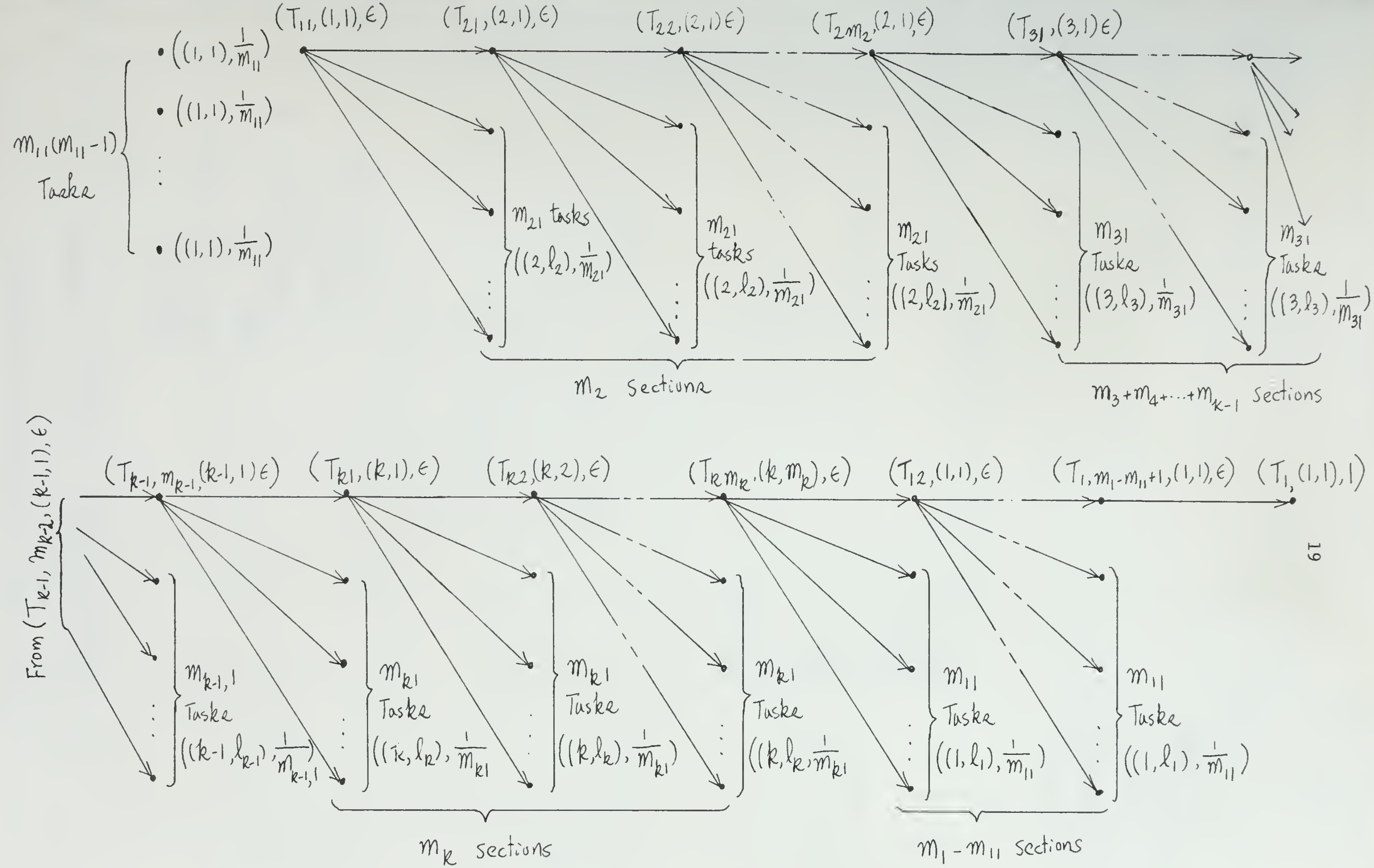


Figure 5a. Precedence Graph For Example 2

(ε is an arbitrarily small constant)

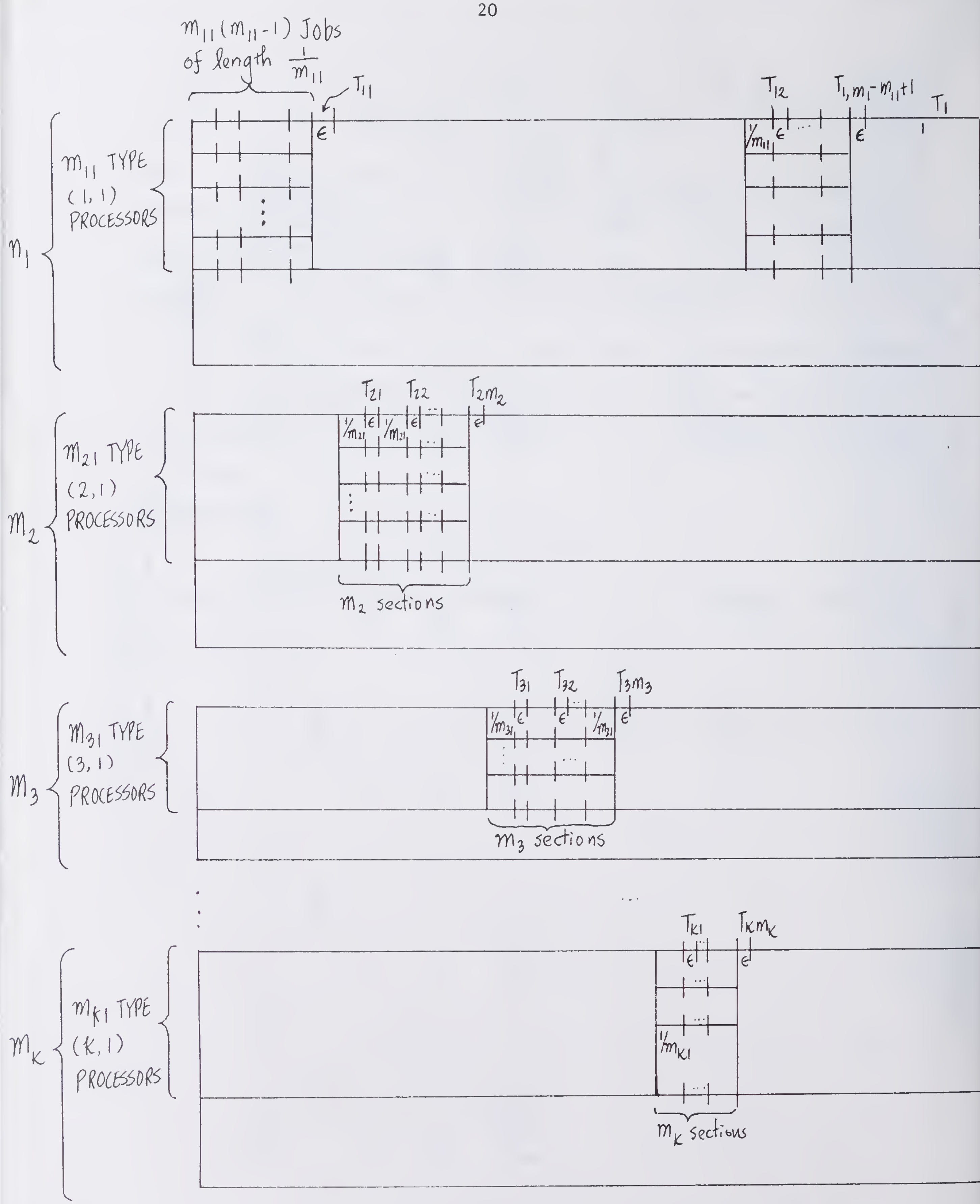


Figure 5b. Worst Case Priority-Driven Schedule

$$\omega = \sum_{d=1}^{k} \frac{m_\ell}{m_{\ell\omega}} + \frac{m_{11}^{-1}}{m_{11}} + (m-m_{11}+2)\ \epsilon;\ m_{11}>m_{21}>m_{k1}$$

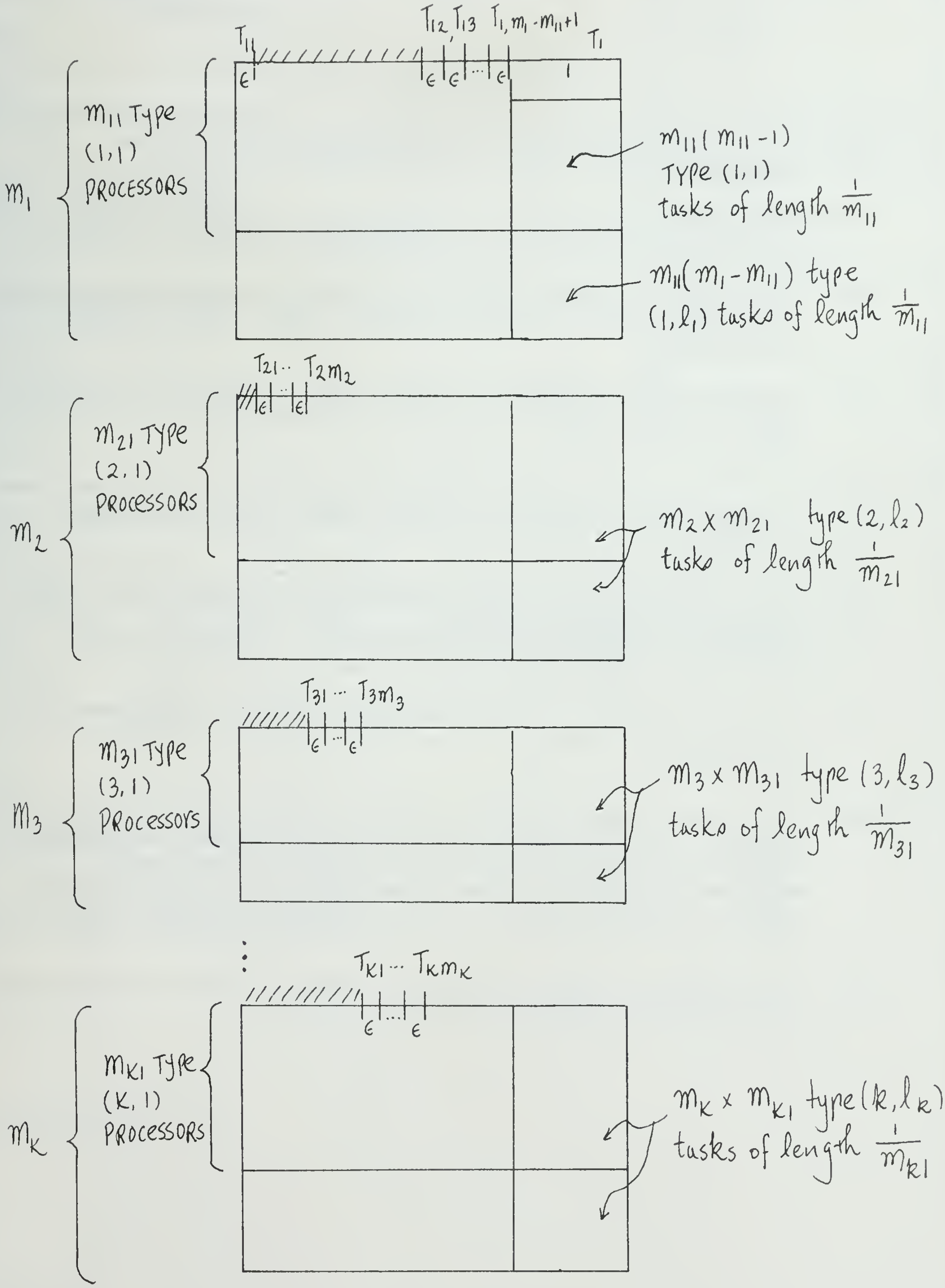


Figure 5c. Optimal Schedule

$$\omega = (m-m_{11}+2)\varepsilon + 1$$

References

[1]. T. C. Hu, Parallel Scheduling and Assembly Line Problems, Oper. Res. 9, No. 6 (1961), pp. 841-848.

[2]. E. G. Coffman, Jr. and R. L. Graham, Optimal Scheduling for Two Processor Systems, Acta Informatica, 1, (1972), pp. 200-213.

[3]. R. R. Muntz and E. G. Coffman, Jr., Preemptive Scheduling of Real Time Tasks on Multiprocessor Systems, J.ACM 17, No. 2, (1970) pp. 324-338.

[4]. J. Bruno, E. G. Coffman, Jr. and R. Sethi, Scheduling Independent Tasks to Reduce Mean Finishing Time, Comm of ACM, 17, No. 7, (1974), pp. 382-387.

[5]. J. W. S. Liu and A. Yang, Optimal Scheduling of Independent Tasks on Heterogeneous Computing Systems, Proc. 1974 ACM National Conference, pp. 38-45.

[6]. R. L. Graham, Bounds on Multiprocessing Timing Anomalies, SIAM Journal on Applied Mathematics, 17, No. 2, (1969) pp. 263-269, also Bounds on Multiprocessing Anomalies and Related Packing Algorithms, AFIPS Conf. Proc., 40 (1972), pp. 205-217.

[7]. J. W. S. Liu and C. L. Liu, Bounds on Scheduling Algorithms for Heterogeneous Computing Systems, IFIP Conf. Proc. (1974) pp. 349-353.

[8]. D. G. Kafura and V. Y Shen, Scheduling Independent Processors with Different Storage Capabilities, Proc. ACM National Conference (1974), pp. 161-166, also K. L. Krause, V. Y. Shen, and H. D. Schwetman, Analysis of Several Task Scheduling Algorithms for a Model of A Multiprogramming Computer System, J. ACM, 22, No. 4, (1975) pp. 522-550.

[9]. G. L. Chesson, Communication and Control in Cluster Network, Proc. ACM National Conference (1974) pp. 509-514.

BIBLIOGRAPHIC DATA SHEET	1. Report No. UIUCDCS-R-77-835	2.	3. Recipient's Accession No.
4. Title and Subtitle Performance Analysis of Multiprocessor Systems Containing Functional Dedicated Processors			5. Report Date May, 1977 6.
7. Author(s) Jane W. S. Liu and Chung L. Liu			8. Performing Organization Rept. No.
9. Performing Organization Name and Address Department of Computer Science University of Illinois at Urbana-Champaing Urbana, IL 61801			10. Project/Task/Work Unit No. 11. Contract/Grant No. MCS-73-03408
12. Sponsoring Organization Name and Address National Science Foundation Washington DC			13. Type of Report & Period Covered 14.

15. Supplementary Notes

16. Abstracts

General models of multiprocessor systems in which processors are functionally dedicated are described. In these models, processors are divided into different types. Clearly, the model of multiprocessor systems with identical processors is a special case of our models. These models also include the job shop problem in which there is exactly one processor of each type. Worst case performance bounds of priority-driven schedules are obtained.

17. Key Words and Document Analysis. 17a. Descriptors

worst case performance, multiprocessors systems, scheduling algorithms, and functionally dedicated processors.

17b. Identifiers/Open-Ended Terms

17c. COSATI Field/Group

18. Availability Statement	19. Security Class (This Report) UNCLASSIFIED	21. No. of Pages
	20. Security Class (This Page UNCLASSIFIED	22. Price

FORM NTIS-35 (10-70) USCOMM-DC 40329-P71